A Microservice Infrastructure for Distributed Communities of Practice

A Microservice Infrastructure For Distributed Communities of Practice

A Microservice Infrastructure For Distributed Communities of Practice

Peter de Lange[1], Bernhard Göschlberger[2,3], Tracie Farrell[4] and Ralf Klamma[1]

[1] RWTH Aachen University
lange,klamma@dbis.rwth-aachen.de
[2] Research Studios Austria FG
[3] Johannes Kepler University Linz
goeschlberger@researchstudio.at
[4] Open University UK
tracie.farrell-frey@open.ac.uk

Abstract. Non-formal learning in Communities of Practice (CoPs) makes up a significant portion of today's knowledge gain. However, only little technological support is tailored specifically towards CoPs and their particular strengths and challenges. Even worse, CoPs often do not possess the resources to host or even develop a software ecosystem to support their activities. In this paper, we describe a distributed, microservice-based Web infrastructure for non-formal learning in CoPs. It mitigates the need for central infrastructures, coordination or facilitation and takes into account the constant change of these communities. As a real use case, we implement an inquiry-based learning application on-top of our infrastructure. Our evaluation results indicate the usefulness of this learning application, which shows promise for future work in the domain of community-hosted, microservice-based Web infrastructures for learning outside of formal settings.

Keywords: learning infrastructures, microservices, communities of practice

1 Introduction

The vast majority of human learning happens outside of formal settings. Learning activities may be quite informal, as found in incidental learning, self-regulated learning and socialization [18]. Some learning may involve more structure or planning, which is generally referred to as non-formal learning [5]. A significant portion of this learning happens in Communities of Practice (CoPs) [20]. These communities are not bound together by an organization, but rather by sharing a common craft or profession, with the desire to learn from each other through knowledge sharing. While only few CoPs have the size and influence to get tools tailored to their needs, the long tail [1] of CoPs does not possess the resources, such as central hosting infrastructures or shared budget. Consequently, they often adopt publicly available tools (e.g. social software) and re-purpose them according to their needs, mitigating the tools' technical shortcomings through socially enforced usage policies. Thereby, the CoP becomes dependent on the tool provider and also loses control over its data. Even if a CoP manages to establish a centralized infrastructure, this often results in dependencies on single, knowledgeable members or institutions and does not account for dynamic membership, a common characteristic of CoPs.

As a consequence, we claim that a suitable infrastructure for CoPs needs to be decentralized and managed by the community members themselves. It should be easily deployable, extensible and flexible in terms of scalability and accessibility from the outside. The microservice paradigm [14] with loosely coupled services bound together by lightweight protocols fits these demands perfectly. Combined with an underlying peer-to-peer (p2p) network of nodes managed by the CoPs themselves, the microservices should self-replicate through the network according to the community's current needs. Once deployed on the infrastructure, those services and development efforts should remain available, even after the contributing member has left the CoP. Like the ship in the Theseus paradox, a community should be able to persist, even though all of its members have changed over time, as long as there are people willing to engage. Serving as a *community's long term memory*, the infrastructure allows members to learn from their "ancestors", much like we can observe in scientific communities. Just like opening the water tap, using a certain learning environment should be available to every community member at all times. Thus, we propose a *Learning as a Utility* approach, which makes it possible for all community members to equally engage in development, hosting and using learning applications.

The contribution of this work is twofold. First, we describe a technical infrastructure that provides CoPs with an independent, sustainable and flexible way of developing, hosting and sharing their state-of-the-art learning applications on the Web. Second, we present a distributed version of a proven method for inquiry-based learning. Following a design science approach as proposed by Hefner [8], we start by presenting a real-world use case (Sec. 2). From this, we derive the functional requirements of the realized application and the technical design of both our infrastructure and application (Sec. 3). We evaluate our designed application in multiple iterations and discuss the implications (Sec. 4), before presenting related work (Sec. 5) and concluding this contribution (Sec. 6).

2 Use Case: Distributed Inquiry-Based Learning

In our use case, a community of young European youth workers are preparing for participation in a European-funded training course on "creative leadership". The participants are an international group, with different levels of experience, from multiple organizations and countries. The team must create learning content that appeals to this diverse group and meets their needs, which is a challenge given the complexity of both creativity and leadership as learning subjects. In addition, the three trainers providing the course are distributed across different countries and organizations as well, with no possibility to meet beforehand. Since the whole CoP neither shares a geographic location, nor central infrastructure or budget, this use case stands exemplary for the needs and challenges of distributed communities of practice.

To help establish the boundaries of the participants' knowledge and identify common ground or potential conflicts, the trainers want to find out which questions the participants have about creative leadership and how those questions relate to one another. Specifically, the trainers implement a form of *Question-Based Dialog* called Noracle [6] before the training starts, to model and visually represent their common space of ignorance about creative leadership. This special form of inquiry-based learning starts with a central question raised by the trainers, which is then answered by the participants by raising follow-up questions. This way, the *Community Ignorance* becomes visible and the trainers gain insight about what the participants are interested in and their views on the subject. As participants create this *Problem Space*, they document the questions that they have about creative leadership, their assessments of the questions that others stated and any links they perceive between them. In its current form, this involves an on-scene session at the start of the training course, where the community has a limited time-frame to establish their community ignorance by writing down questions they have. A digital version of the concept could be applied already before the community meets. We state the following research questions:

R1: Does a digital version affect the community's perception of their ignorance?
R2: Can a decentralized learning infrastructure be managed by the community?

3 Realization of the Distributed Noracle

In this section, we describe the realization of a digital and distributed version of the Noracle method, an application which we first envisioned in [4]. It fulfills the use case described in the previous section and makes it possible to explore and map community ignorance through question-based dialog, asynchronously and without a formal infrastructure. A space is the main view of the application (shown in Fig. 1). Users can create a space and invite others to the space by sharing an invitation link. The user interface provides a list of subscribed spaces such that users can switch between spaces with two clicks. The space view consists of a canvas displaying the questions and their relations as a graph of speech bubbles. It also features a list of users subscribed to the space and a (collapsible) help section. Below the canvas, users can select their current interaction mode. The "Select/Navigate" mode allows users to define the portion of the graph that is displayed. Selected questions and direct neighbors of selected questions are displayed. If a displayed question that is not yet selected has neighbors that would be displayed upon selecting it, they are symbolically indicated as additional speech bubbles behind the question. In the "Drag and Zoom" mode, users can move questions around freely, as well as pan and zoom, to either view parts of the graph in detail or get a birds eye view. The "Add Question" and "Add Relation" mode allows users to add questions or relations by clicking on one question (add a question) or two questions (add a relation).

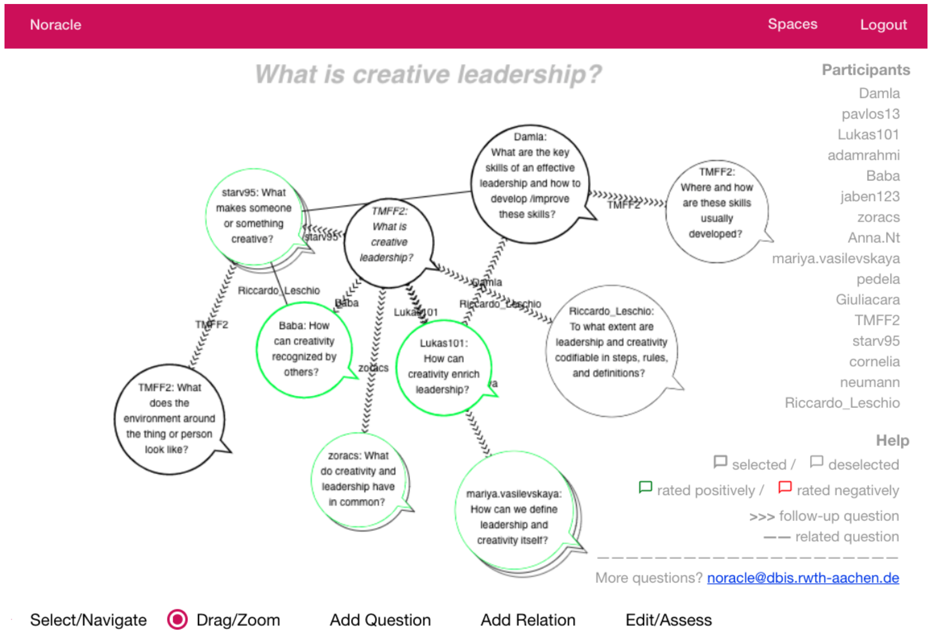

Fig. 1. Screenshot of the Distributed Noracle application

Then, a dialog window opens that asks the user to enter the text of the question or the type of the relation. For relations, we allow for both *Follow Up* relations (depicted as small arrows indicating the direction), which is the default type of relation that is created between a new question and its parent question, as well as *Link* relations (depicted as straight lines) that display a certain connection of similar questions, although they are not in a direct *Follow Up* relationship. Finally, the "Edit/Assess" mode enables users to either modify their own questions and relations or to assess the value of questions or relations of others. We use a coloring mechanism that displays the entity according to its overall rated usefulness in a specific color, ranging from green to red.

Fig. 2 shows an exemplary usage scenario of a Distributed Noracle session. While *Bob*'s node features the set of microservices that realize the application, *Alice* has decided to start an empty node without any services running on it. This can have several reasons, also including the lack of resources, both in terms of computing power or, especially in mobile settings, energy. *Carol*'s node also contains a set of Noracle microservices, whilst *Dave* has not started a node at all and uses *Bob*'s node to access the remote Web frontend for participating in the collaborative session. As this scenario demonstrates, our framework provides flexible access to the application with several possibilities to join a session. Depending on the currently available resources of a community member, our framework allows to flexibly start and stop (parts of) applications on a node. Because a central infrastructure is unavailable, this usage scenario does not fea-

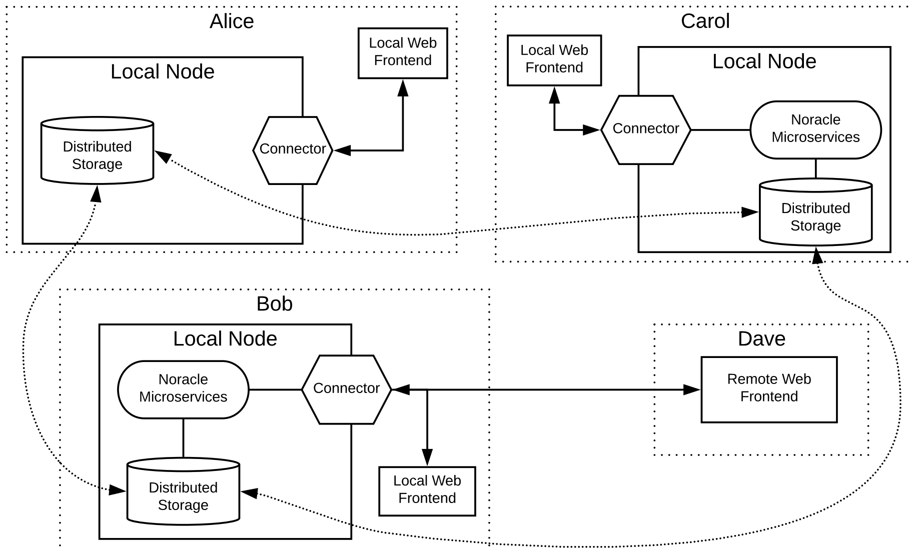

Fig. 2. Exemplary usage scenario of the Distributed Noracle

ture any centralized component, like a master node or a central URL for the Web frontend. Rather, the whole infrastructure is distributed among the community. In the following, we first present a short overview of our technical infrastructure, before we describe the realization of the Distributed Noracle in more detail.

3.1 A Distributed Microservice Infrastructure

The technical basis we use for this work is called las2peer [10], an open source p2p framework for implementing and hosting Java microservices. Every las2peer node in our distributed community learning infrastructure consists of at least two components. The first is the *Distributed Storage*. This storage is partitioned and partly duplicated throughout the network, allowing for a shared, yet synchronized data store. Technically, we base our storage and inter-node communication mechanisms on the FreePastry library[1], a p2p overlay network that provides both a messaging system as well as a *DHT* (Distributed Hash Table) storage system. To ensure privacy, security and data protection, we added end-to-end encryption in form of an *Envelope* system on top of it, ensuring each message and all data stored via the system is encrypted. The second component a node has to integrate is the so called *RESTful Web Connector*. It realizes the communication to the outside, with the capability of routing RESTful calls to an application's (Gateway) interface.

Our framework is capable of load balancing requests to microservices in the entire network, may it be because the service simply does not exist on the local

[1] `http://www.freepastry.org`

node, or the node is currently overloaded with requests and offloads the task to other nodes in the network. Upstarting services register themselves to the network by calling a specific routine of the node, which then manages their location in the shared storage for all nodes to look-up. This *Sidecar Pattern*-like service registration and discovery ensures that a connector will find the nearest service that currently is flagged as being capable of taking requests. The communication between microservices is realized using a *Message Oriented Middleware* (MOM) that is based on a *Publish & Subscribe Pattern*. Each node registers all running services as subscribers to their corresponding "Service Topic". If a service wants to call another service, it performs a remote method invocation that is sent throughout the network. A node hosting a corresponding service that receives this request will route it to the service, which will handle it. The answer is then sent again in the same way throughout the network. Several timeout mechanisms and an acknowledgment system prevent messages with missing receiver to be forwarded endlessly or messages being answered by multiple services. By using the p2p network to enforce an *Event-driven Architecture* (EDA) of microservice-based applications, we target the needs of fast-changing topologies in CoPs, where complete knowledge of the network might both not be available or even desirable. Nodes can join and leave the network at any time, and the network keeps a persistent shared storage with *Eventual Consistency* (following the BASE model of modern cloud computing architectures [16]), regardless of the current topology. Besides this, it is of course possible for a microservice to implement and maintain its own database, separately of the distributed storage.

3.2 Building the Distributed Noracle

The Distributed Noracle application consists of a set of five microservices. A *Space Service* handles the creation of spaces and their members. The *Question Service* takes care of creating and updating questions, while the *Relation Service* does the same for relations. The *Vote Service* handles both votes for questions and relations. Finally, the *Agent Metadata Service* is responsible for storing additional metadata (such as the name) for the members of the CoP. Additionally to these five services, the *Noracle Service* serves as the *Gateway Service* of the application. It differentiates itself from the other microservices that make up the application by providing a RESTful API to the outside. Apart from this, it is implemented as any other microservice in the network, the difference is in terms of semantics (e.g. it does not access the distributed storage facilities). Being called by the connector, it distributes the requests to the set of microservices we just described.

To give a concrete example of inter-microservice communication of the Distributed Noracle application, consider an incoming request for creating a question. This RESTful request would be transferred from the *RESTful Web Connector* to the *Noracle (Gateway) Service*, which would send a request to the *Question Service*. This service in term would invoke the corresponding *Space Service* for further details, for example if the user is allowed to create a question in this particular space. Upon receiving the answer from the *Space Service*, the

Question Service would create a new *Question* object in the distributed storage and call the *Relation Service* for creating the corresponding relation between the newly created question and its parent. Finally, the *Question Service* would answer to the *Noracle (Gateway) Service* so that it can forward the HTTP Response to the *Web Frontend*, whether the question has been successfully created. This particular scenario is not necessarily limited to a single node, the microservices can be situated anywhere in the network and it is also neither needed nor desired that a particular microservice knows which instance of the called microservice did handle the request. In the exemplary usage scenario depicted in Fig. 2, if Alice's node receives such a request, it would be distributed throughout the network, because Alice's node does not host any of the application's microservices. Depending on their current load, the request would be processed by the node of either Carol or Bob, and their *Noracle (Gateway) Service* would possibly distribute the just described sub-request again to microservices on other nodes. The flexible scalability of the infrastructure also allows several instances of the same microservice residing at a node, spawning automatically according to the current need. The infrastructure is designed for failure in a way, that non-responding microservices are automatically shut-down and replaced by new instances.

The frontend of our application is based on the Angular 4 framework and it is part of the node, served from the distributed storage. Therefore, we developed a *File Service* that provides a RESTful interface for storing and serving Web frontends directly from the network, removing the need for an additional Web server. Authentication is done using the OpenId Connect *Single Sign-on* (SSO) standard. To provide CoP members with the software needed to start their own node, we created a *Node Package*. It is a small folder that contains an empty node preconfigured to connect to a network via a (configurable) *Seed Node*. It then replicates the microservices of the application via the p2p network and starts them locally. The application and its underlying framework are released as open source software[2].

4 Evaluation

We evaluated our application in four iterations, including one preliminary evaluation, with different types of learning communities. Each evaluation had a certain focus that lead to a gradual improvement of the tool. In the following, we describe each of these evaluation in more detail.

4.1 Preliminary Evaluation

In the preliminary evaluation, a Web science research group at a university used a paper mock-up of the Distributed Noracle for questioning current priorities in their research field. The purpose of this evaluation was to determine whether

[2] `https://distributed-noracle.github.io`

the method could be transferred to a digital space and which features would be required. This community was appropriate because of the shared interest in a topic, diverse levels of experience, and a loose collaborative structure.

Participants and Procedure: 8 members of the community took part in the trial. Half of the participants were more experienced members of the team, as determined by whether or not they were supervising PhD students. The other half were PhD candidates or post-doctoral researchers. To represent a shared digital space, the participants worked asynchronously on a large poster in the lab. A general reflection question was posed as the central question in the Distributed Noracle mock-up: "What is the most relevant, open question for social semantics?" Each participant received a differently colored marker to represent her contributions to the poster. As participants added questions, they were also asked to circle questions they supported and draw links between questions to show their relationship. Participants also starred those contributions they thought were most helpful. The evaluation lasted for three days.

Analysis and Outcomes: After concluding the exercise, the participants completed a short evaluation on the insights they could draw from looking at the question graph. They also expressed thoughts about the overall value of the proposed artifact. The main outcome of this evaluation was that the tool could help to *structure dialog more efficiently and encourage users to consider broader or new perspectives*, but that *participants need assistance in interpreting the graph.* The need to transfer the process of question-based dialog to a digital space to increase its value was established through this evaluation.

4.2 Interface Evaluation

The first evaluation of the digital tool was conducted with participants on an "on arrival" training for participation in the European Voluntary Service (EVS) program. The participants used the Distributed Noracle to consider the future of European youth work in the context of a project planning session. This community was appropriate because of the ill-defined nature of the topics that participants were exploring and the lack of shared infrastructure between them.

Participants and Procedure: 7 participants between the age of 20-25 from different European and Erasmus+ partner countries took part in the study. The participants had similar levels of experience in the area of youth work (1-2 years). In this evaluation, the participants worked synchronously. All participants used a given link to access the single-node deployment of the Distributed Noracle. After a project planning session in their face-to-face seminar, the participants joined the space and continued their reflections online. They had a set period of time to explore the application with the general reflection question posed to them: "What is the future of European Youth Work?" As participants added questions, they were also asked to assess questions they found helpful and create links between different questions to show their relationship. The exercise lasted for approximately 30 minutes.

Analysis and Outcomes: The addition of some analytic features helped users to get a sense for a question's *importance, quality and validity.* Examples for this are the marking of questions where conflicts are present in red, or darkening the circle that surrounds the topic as more and more contributors agree that the question is relevant. Users made suggestions primarily for improvements related to the interface, as some participants found the layout and animations slightly disorientating. This was mainly due to the prototypical nature of the first iteration and we improved the overall look and feel for the next evaluations.

4.3 Technical Evaluation

The second evaluation was conducted with workshop participants of the Joint European Summer School on Technology Enhanced Learning (JTELSS). The purpose of this evaluation was to test the technical features of the tool, in particular the distributed architecture. The community was considered appropriate for a technical evaluation of the learning tool because of their experience with educational software.

Participants and Procedure: Approximately 20 people participated in the workshop. First, the participants were given a short introduction to the method of question-based dialog and to the application. As part of this introduction, participants were guided on how to start their own node and join the network. Participants used their own technical devices to launch their nodes. We provided a local seed node the participants could connect to. The participants were then given about 20 minutes of time to explore the tool. We provided a general starting question in a sample space. Participants were also asked to assess questions they found helpful and create links between different questions to show their relationship. In addition, they were invited to create their own space and invite other participants to join.

Analysis and Outcomes: Despite some technical problems, mainly related to firewall restrictions of the local WiFi network, most of the participants were eventually able to connect their node to our on-scene network. Participants not able to start their own node used other participant's nodes to join the problem space, and thus were able to participate as well. This proved the capability of starting ad-hoc Distributed Noracle networks within a community. The data we received from this evaluation was afterwards used to improve the application, leading to a more stable version used in our pedagogical evaluation.

4.4 Real-World Pedagogical Usage Evaluation

The third evaluation was conducted with the community described in Section 2. Participants of an European training course on creative leadership were invited to participate in an experiment using the Digital Noracle to help prepare for the course and get a sense of the participants' existing knowledge gaps. The purpose of this evaluation was to test the application in a real asynchronous and distributed setting, adding monitoring data to the qualitative verbal and written data.

Participants and Procedure: 34 participants took part in the evaluation. The participant group was diverse, with different nationalities, levels of experience and knowledge about the subject of the training course, *Creative Leadership*. One week before the training course, participants were notified via email that an "experiment" would be taking place, using a beta version of an application to help prepare for the training. They were informed that their participation in the experiment was completely voluntary, but that it would help to establish what participants found most confusing or difficult about the concept of creative leadership. They received information on how to join the Distributed Noracle and were invited to contribute questions to a specific reflection question related to the training course. Since the participants were locally distributed with prior contact only via email, we created an artificial distributed setting by creating a network of nodes at a university. We provided a URL to the participants that automatically distributed them to their specific node. This created a scenario where each participant had her own node, without the actual need for a technical setup procedure that would have been unfeasible for this particular evaluation, especially regarding the evaluation of the results. After the first 48 hours, participants were asked via email to review the questions that other participants had posted so far once again and evaluate how important or useful they are to the over-all discussion. Once the participants arrived at the training course, the entire trainer team and the trainees participated in an analysis of the question graph and an evaluation of the tool's features. The evaluation included three items: *What insights can you draw from the graph? What features or functions might improve the value of this tool for you? In which situations could you imagine to use it?* Each individual had five minutes to review the graph and to take some notes. Then, the facilitator gathered the insights in a plenary session, during which the participants' statements were also clustered according to their shared theme.

Analysis and Outcomes: With regard to the insights that could be drawn from the graph, the group found it quite easy to see what is important, such as focusing on the development of creative skills. They noticed that many questions related to this topic in some way. There was a considerable agreement about the importance of these types of questions (as indicated by the green color). They also realized that they had taken a *very individualistic perspective on creativity and leadership*, with very few questions having to do with the social aspect of creative development. This type of reflection can be mainly contributed to the graph-like structure of the problem space, with its highlighting of importance capabilities. The way that questions were formulated allowed the participants to differentiate between questions related to defining creativity and questions related to the process of developing or improving creativity. Features that participants felt were important to develop had to do with analytic features to help uncover other types of insights or consequences. For example, only one trainee had noticed that similar questions were repeated several times in the graph. In addition, a third of the participants said that they would find it helpful if there was a way of knowing exactly how many people or a percentage of people found a question useful. All of the participants and the trainer team felt that the tool would be improved by having a way of visualizing what insights or consequences could be drawn. The trainees agreed that the tool helped establishing the interests of a group in advance, which is useful in a variety of settings. The training team remarked, that instructions were extremely important in helping the participants to know how to use the application. Especially with new users, facilitation could be very useful in helping to maintain the quality of the space by demonstrating question-asking and some of the application's additional features. The training expressed the usefulness of the application as a preparatory exercise for a training course, workshop or seminar.

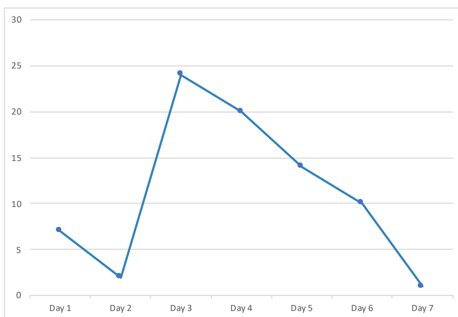

Fig. 3. Activities over time

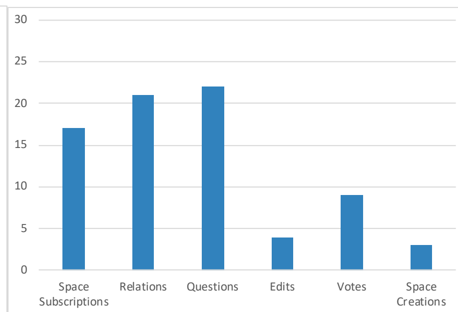

Fig. 4. Activities by type

Additionally to our previous evaluations, we monitored the complete network for user activities [17]. Fig. 3 shows the relevant activities monitored during the one week period we had the network running for this evaluation, while Fig. 4 shows the complete number of (selected) monitored activities per type. We started the monitoring the day we sent out the invitation mail, while we asked the participants to start their 48h collaboration phase on the beginning of day three. As one can see, activity is high between day 3 and 5, while it declines afterwards. Still, the number of recorded activity before and after this "official" trial phase shows the intrinsic motivation participants had to visit the problem space, an important factor for learning activities in self-regulated learning scenarios. Another interesting observation we made during analyzing the monitoring data was, that the average question depth was 1.9, meaning that on average a question was about two questions away from the seed question. We perceive this as another indicator of the usefulness of the graph-based visualization, since most questions did not connect directly to the seed question, but to follow-up questions, demonstrating the evolving awareness of the community ignorance, represented by the growth of the graph.

4.5 Discussion

Improvements proposed by users mostly dealt with the interface and analytic features, such as additional ways of visualizing other aspects of the dialog by making nodes larger or smaller, allowing for certain questions to be marked as "resolved" and additional ways of linking questions. Most of the users in all three evaluations said that such a tool can be useful in the planning stages of a project and at the beginning of any complex task or assignment to gain orientation. In addition, participants saw affordances for structuring group- and teamwork in schools.

The trainer team of the real-world pedagogical usage evaluation stated they were able to save considerable time in gathering important information on the trainees' expectations and knowledge. In a typical training scenario, a half day would have been spent on these types of abstract questions about the program. In this case, it only took 45 minutes of analyzing the resulting question-graph to achieve an even better result. In addition, starting the process in advance seemed to have the effect that the group took the exercise more seriously, which lead to these better results. Possible reasons for this mentioned by the trainers were that when the method is used in face-to-face settings, the participants are naturally distracted by the person they have in front of them. The tendency to move towards providing answers or advice makes it more difficult to keep them on task. Working asynchronously with the participants appeared to have resolved this as it was not necessary to always repeat that the participants should only ask questions.

From the technical point of view, due to their prototypical nature, the evaluations showed potential weak points of our application, such as the stability and ease of starting a node. While we were able to solve many technical challenges during and after the technical and pedagogical evaluation, we are still working on improving both points. Nevertheless, all three different evaluation scenarios proved that our prototype is already applicable in real-world usage scenarios.

5 Related Work

Question asking is seen as one of most important skills for innovation, since it contributes to lateral thinking and thus better problem solving [19]. Question-based dialog is viewed as a specific type of a sense-making tool that is also *represention-centric* [11]. To help structure discourse analysis, computational linguistics has offered frameworks to examine collaborative sense-making in virtual environments [9]. For example, argumentation platforms offer a representation-centric approach to collaboration. Contributions are visually represented, categorized as issues, claims, premises and evidence, with modifying functions to support or refute other constituents of the argument. Cohesion graphs of discussion threads, which represent contributions as nodes at different levels, can examine lexical chains in discourse analysis to understand influence on conversation and identify key issues in conversation. Related works in this domain mostly deal with the issue of how face-to-face scenarios differ from online discussions and how to aggregate community knowledge [12]. Instead of representing *knowledge* in the form of arguments, the Distributed Noracle examines the *gaps* in community knowledge in the form of questions.

The question of system maturity, flexibility and also interoperability is still an active research area [15]. The idea of using p2p-based systems for sharing of educational resources came up first with the creation of EDUTELLA [13], a network for exchanging information about learning objects. Recent development in this area is the InterPlanetary File System [3] project, which describes itself as a *peer-to-peer hypermedia protocol* and shares the concern for *increasing consolidation of control [on the Web]*. Related development approaches have been characterized as p2p cloud computing [2] and edge-centric computing [7]. Despite the high research activity in this domain, we did not find any recent approaches that focus on supporting CoPs with self-managed, decentralized infrastructure. Forums, blogs and wikis are still the most commonly adopted tools for CoPs that need to accommodate geographically distributed participants at scale. However, they do not preserve a representation of contributions that can be elaborated or amended as the community changes, making them harder to sustain for CoPs.

6 Conclusion and Future Work

In this paper, we presented both a microservice-based Web infrastructure for distributed learning communities and an application of it in form of an inquiry-based learning tool for CoPs. We followed a design science approach and incrementally tailored our application to the needs of the community, according to the outcome of each evaluation. Our approach concentrated on taking into account the specific attributes of CoPs, like temporal and spatial dynamics. By consequently addressing these attributes, we support CoPs in their efforts to share and acquire knowledge. As information remains available throughout the communities' existence and services evolve continuously at the same time, our infrastructure ensures sustainability and adaptability, aptitudes we reckon to be crucial in the development of a more democratic and egalitarian Web.

In future work, we want to improve our distributed monitoring by ways of providing this information to the community. One particular approach we are working on is the introduction of social learning bots that guide the users through the problem space, tailoring themselves to the user by analyzing the previously monitored usage data. Furthermore, a feature for checking similar questions and also tracking how often they arise could be useful. We are also working on a way how to visualize if a question has been resolved. Finally, we are investigating ways of improving the underlying framework to be even more easily manageable by CoPs. In particular, the switch from the microservice paradigm to a "serverless", Function as a Service (FaaS) supporting platform seems worth investigating.

Acknowledgments. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 687669 (WEKIT) and from the European Unions Erasmus Plus programme, grant agreement 2017-1-NO01-KA203-034192.

References

1. Anderson, C.: The Long Tail: Why the Future of Business Is Selling Less of More. Hyperion (2006)
2. Babaoglu, O., Marzolla, M.: The people's cloud. IEEE Spectrum 51(10), 50–55 (2014)
3. Benet, J.: IPFS - content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561 (2014)
4. de Lange, P., Farell-Frey, T., Göschlberger, B., Klamma, R.: Transferring a question-based dialog framework to a distributed architecture. In: EC-TEL'17. pp. 549–551. Springer (2017)
5. Eshach, H.: Bridging in-school and out-of-school learning: Formal, non-formal, and informal education. Journal of science education and technology 16(2), 171–190 (2007)
6. Farrell-Frey, T., Gkotsis, G., Mikroyannidis, A.: Are you thinking what i'm thinking? representing metacognition with question-based dialogue. In: ARTEL'16. pp. 51–58 (2016), http://ceur-ws.org/Vol-1736/
7. Garcia Lopez, P., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., Barcellos, M., Felber, P., Riviere, E.: Edge-centric computing: Vision and challenges. SIGCOMM Comput. Commun. Rev. 45(5), 37–42 (2015)
8. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. 28(1), 75–105 (2004)
9. Iandoli, L., Quinto, I., De Liddo, A., Buckingham Shum, S.: On online collaboration and construction of shared knowledge: Assessing mediation capability in computer supported argument visualization tools. JASIST 67(5), 1052–1067 (2016)
10. Klamma, R., Renzel, D., de Lange, P., Janßen, H.: las2peer - a primer. ResearchGate (2016), https://dx.doi.org/10.13140/RG.2.2.31456.48645
11. McLoughlin, C., Patel, K., O'Callaghan, T., Reeves, S.: The use of virtual communities of practice to improve interprofessional collaboration and education: findings from an integrated review. J. Interprof. Care 32(2), 136–142 (2018)
12. Meyer, K.A.: Face-to-face versus threaded discussions: The role of time and higher-order thinking. JALN 7(3), 55–65 (2003)
13. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., Risch, T.: EDUTELLA: A p2p networking infrastructure based on RDF. In: WWW '02. pp. 604–615. ACM (2002)
14. Newman, S.: Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, Inc. (2015)
15. Ochoa, X., Ternier, S.: Technical learning infrastructure, interoperability and standards. In: Technology Enhanced Learning, pp. 145–155. Springer (2017)
16. Pritchett, D.: BASE: An acid alternative. Queue 6(3), 48–55 (2008)
17. Renzel, D., Klamma, R., Jarke, M.: IS success awareness in community-oriented design science research. In: DESRIST'15. pp. 413–420. Springer (2015)
18. Schugurensky, D.: The forms of informal learning: Towards a conceptualization of the field. Tech. rep., Centre for the Study of Education and Work (2000)
19. Sloane, P.: The Leader's Guide to Lateral Thinking Skills: Unlock the Creativity and Innovation in You and Your Team. Kogan Page (2017)
20. Wenger, E.: Communities of Practice: Learning, Meaning, and Identity. Learning in doing, Cambridge University Press, Cambridge, UK (1998)